COMPARATIVE ANALYSIS OF K-MEANS DATA MINING AND OUTLIER DETECTION APPROACH FOR NETWORK-BASED INTRUSION DETECTION

[1] *Lazarus Kwao, MPhil IT, Dept. Of Computer Science, KNUST, Kumasi, Ghana, lazoe16@yahoo.com.*
[2] *Joseph Kobina Panford, Dept. Of Computer Science, KNUST, Kumasi, Ghana, jpanford@yahoo.com*
[3] *James Ben Hayfron-Acquah, Dept. Of Computer Science, KNUST, Kumasi, Ghana, jbha@yahoo.com*

Abstract - New kind of intrusions causes deviation in the normal behaviour of traffic flow in computer networks every day. This study focused on enhancing the learning capabilities of IDS to detect the anomalies present in a network traffic flow by comparing the k-means approach of data mining for intrusion detection and the outlier detection approach. The k-means approach uses clustering mechanisms to group the traffic flow data into normal and abnormal clusters. Outlier detection calculates an outlier score (neighbourhood outlier factor (NOF)) for each flow record, whose value decides whether a traffic flow is normal or abnormal. These two methods were then compared in terms of various performance metrics and the amount of computer resources consumed by them. Overall, k-means was more accurate and precise and has better classification rate than outlier detection in intrusion detection using traffic flows. This will help systems administrators in their choice of IDS.

***Key Words*:** K-Means, Outlier Detection Approach, Intrusion Detection, Network- based, NOF, clusters

1. INTRODUCTION

1.1 Background of Study

An intrusion is a malicious or unauthorized attempt or activity to access, modify, control, or create an unreliable or unusable system (Anderson, 1980, and Jirapummin, 2000). Intrusions attempt, or intrusions can result from external or internal intruders. Today, it is difficult to maintain a high level of security to ensure secure and reliable communication of information between different organizations as the speed and complexity of networks increases rapidly, especially when these networks are open to the public. The number and types of intrusions have increased considerably. Secure communication via computer networks and other systems carries the risk of intrusion and abuse. Thus, intrusion detection systems (IDS) have become a necessary part of the security of computers and networks (Hoque, Mukit, Bikas and Naser, 2012). Intrusion Detection Systems addresses three critical security functions: detection, observation and response to illegal actions by intruders. The intrusion detection system (IDS) is used as the second defense in the computer and in the network system to ensure security (Bijone, 2016). An intrusion detection system does not prevent an interruption, detects, observes and informs a system administrator. This response typically incorporates attempts to contain or maintain such damage, for example, when closing a network connection. When an IDS detects illegal system activity, it logs these events, stores important data, activities, alerts, and the administrator through a warning and, in some cases, attempts to intervene. In addition to the undeniable benefits of an IDS, the archived data and recordings provide satisfactory scientific evidence and can be used as evidence in a legitimate legal case against the intruder.

https://sites.google.com/site/ijcsis/
ISSN 1947-5500

1.2 Statement of the Problem

There are intruders trying to get unauthorized access to network systems day after day (Sundaram, 1996). Meanwhile, there are problems such as identifying new attacks when it comes to intrusions entering the network when a large amount of data is available. Therefore, adequate training is needed for the IDS to know new types of attacks very frequently. This study proposes a new technique that detects and reduces the amount of time and resources required by the learning algorith in a network traffic flow, comparing the effectiveness of the k-means approach data mining for intrusion detection and outlier detection using the nearest neighborhood factor.

1.3 Research Objectives

The performance of these two approaches is compared across multiple metrics of confusion and performance metrics and an analysis is performed to determine which of the two approaches is most effective for intrusion detection in network traffic flows.

The anomalies present in the traffic flow are of following types:

- Protocol anomaly (e.g., HTTP traffic on a non-standard port).
- Statistical anomaly (e.g., too much UDP compared to TCP traffic).
- Hybrid anomaly (combination of the above).

1.4 Research Questions

The following are the research questions that were posed to accomplish the objectives.

- Using the performance metrics which of the two approaches is effective is Intrusion detection?
- What is the amount of computer resources consumed by the two approaches?

1.5 Significance of the Study

This This study will be significant in providing an in depth understanding of the performance of the k-means data mining algorithm and outlier detection for intrusion detection. The study will also compare the amount of computer resources (CPU and RAM) consumed by outlier detection and k-means and come out with the one which is time expensive.

2. LITERATURE REVIEW

The extremely connected computing world has equipped intruders and hackers with new techniques for unauthorized activities. The cost of such temporary or permanent damages caused by their activities to computer systems have entreated organizations to increasingly implement several structures to monitor information flow in their networks (Chandel, 2017). Several security methods have been developed to counter these security threats at different levels of the Transport Control Protocol/ Internet Protocol (TCP/IP) protocol stack. These security threats in the form of intrusions are generally hidden in nature and enter the network through packets or flows. To counter such threats, an intrusion detection system is required to alert the network administrators of a possible attack. There are two fundamental strategies to the planning of IDSs. In an exceedingly misuse detection-based IDS, intrusions are detected by examining and exploring events that correspond to established signatures of intrusions or vulnerabilities.

https://sites.google.com/site/ijcsis/
ISSN 1947-5500

Besides, an anomaly detection based mostly IDS detects intrusions by observing for unusual network traffic.

Related Works

K-Means in Intrusion Detection

This major work is done in the areas such as usage of k-means to partition the data, categorizing botnets using k-means, usage of k-means in detecting intrusions in networks.

Bohara, Thakore and Sanders (2016) agreed on a method to carry out intrusion detection using k-means to partition the data as unsupervised learning approach. They have proposed new distance metrics which can be used in the k-means algorithm to carefully relate the clusters. They have partitioned data into more than one cluster and correlated them with known behaviour for evaluation. Their results have proven that k-means clustering is a better method to categorizing the data using unsupervised techniques for intrusion detection when several types of datasets are available.

As clustering algorithm proves to be very beneficial having large unlabelled dataset, Raykov, et al. (2016) provided an entire analysis of the NSL-KDD dataset and the attacks provided. They used k-means algorithm for this purpose and additionally represented the distribution of instances in clusters imparting better illustrations of the instances and making it clearer to apprehend.

In (Lisehroodi, Muda & Yassin. 2013), they proposed a hybrid framework based totally on neural community Multilayer perceptron (MLP) and K-means Clustering.

Wang Shunye et. al (2013) proposed enhanced k-means clustering algorithm basically consists of three steps. The first step talks about the construction of the dissimilarity matrix. Secondly, Huffman algorithm is used to create a Huffman tree according to dissimilarity matrix. The output of Huffman tree gives the initial centroids. Finally, the k-means clustering algorithm is be appropriate to initial centroids to get k cluster as output. Wine and Iris datasets are selected from UIC machine learning repository to test the enhanced algorithm. Proposed algorithm gives better accuracy rates and results than the traditional k-means clustering algorithm.

Md.SohrabMahmud et. al (2012) proposed an algorithm uses heuristic method to calculate initial k centroids. The proposed algorithm yields accurate clusters in lesser computational time. The proposed algorithm initially calculates the average score of each data objects that has multiple attributes and weight factor. Next, the Merge sort is applied to arrange the output that was generated in first phase. The data points are then divided into k cluster. Finally, the nearest possible data point of the mean is taken as initial centroid. Although the proposed algorithm still deals with the problem of assigning number of desired k-cluster as input.

Juntao Wang and Xiaolog (2011) in his study, an improved k-means clustering algorithm to deal with the problem of outlier detection of traditional k-means clustering algorithm. The enhanced algorithm makes use of noise data filter to deal with this problem. Outliers can be detected and removed by using Density based outlier detection method. The purpose of this method is that the outliers may not be engaged in computation of initial cluster centres. The Factors used to test are clustering time and clustering accuracy. The drawback of the enhanced k-means clustering algorithm is that while dealing with large scale data sets, it takes more time to produce the results.

Munz, Li and Carle (2007) proposed a new method for data mining the use of k-means to sense intrusions. They have categorised the flow of data into clusters of normal and abnormal behaviour and have offered well-described facts of the intrusions and data flow.

https://sites.google.com/site/ijcsis/
ISSN 1947-5500

However, researchers have additionally proposed few adjustments in the k-means algorithm to make it adaptable to the type of datasets with unique kind of networks. This proposal of k-means algorithm offered, makes it appropriate to be referenced for studies related to k-means.

Outliers in Intrusion Detection

Chandola., et, al (2009) defines outlier mining as the identification of items, events or observations which do not conform to an expected pattern or other items in a dataset.

Jabez& Muthukumar (2015) explained a new concept for intrusion detection in computer networks using distance and density-based approaches. They have introduced a metrics called as neighborhood outlier factor which is used to measure the anomaly dataset.

Manandhar and Aung (2014) have proposed an anomaly-based IDS using outlier detection. They have used the normal data instances to build a base model and declare all other data instances which do not obeys the base model.

Gosavi and Wadne (2014) compares various unsupervised outlier detection approaches using various techniques of outlier detection such as local outlier factor (LOF) and local distance-based outlier factor (LDOF).

Kriegel, Kroger and Zimek (2010) has focused on few other approaches to outlier detection which includes model-based approach, proximity-based approach and angle based approach for intrusion detection.

Wu et al, (2007) proposed a new outlier mining algorithm based on index tree, named TreeOut, designed to detect the outliers. Outliers have the weight greater than the threshold. in this technique the upper and lower bound of the weight of each record is calculated for r-region and index tree to avoid needless distance calculation. This algorithm is straightforward to implement, and more appropriate to detect intrusions in the audit data. The outlier detection method is effective in reducing false positive rate with desirable and correct detection rate (Bhuyan, et. al., 2011). This work for the generation of outliers for detecting intrusions is as follows.

3. METHODOLOGY

The methodology for this research was Design Science Research using simulation of scenarios. To help with this simulation, Graphical NetworkTrafficView and Wireshark was employed and the architecture for the simulation scenarios is illustrated in Figure1. Graphical NetworkTrafficView and Wireshark were chosen because they have user- friendly Graphical User Interface (GUI) and enables users to capture live network traffic packets in real time and displays general statistics about your network traffic. MATLAB was used for the analysis of the data captured by NetworkTrafficView and Wireshark. Matlab was used to do the performance analysis of the k-means and outlier detection approaches.

https://sites.google.com/site/ijcsis/
ISSN 1947-5500

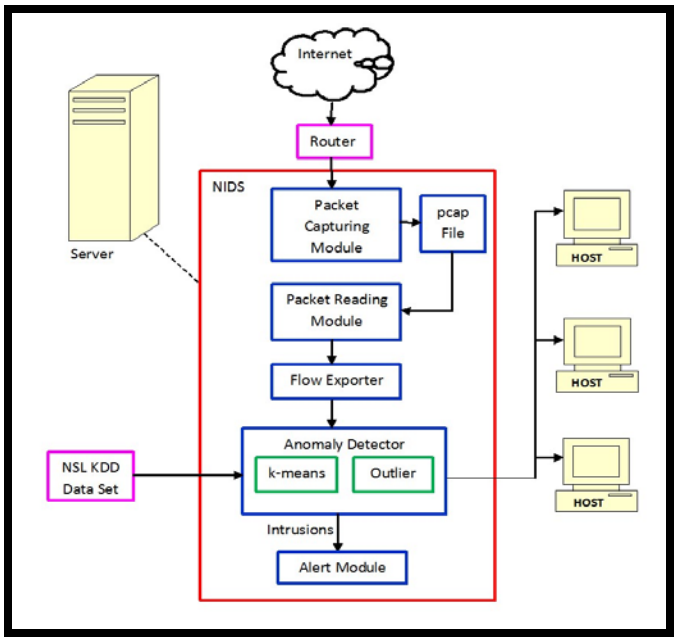

Figure 1: System architecture of IDS (Chandel, 2017)

Input data

The process starts with the collection of input data and clustering them into flow records by the IDS. This input data containing real internet traffic is collected at four times with different types of traffic characteristics. This input data was captured from the network setup of few tertiary institutions connected to each other to share knowledge and is used by students of the tertiary institutions to access the internet. Each set of data captured forms a dataset consisting of many flow records. The study has captured a smaller set of traffic ranging between 1-3 minutes for each dataset for analysis. Each dataset consists of varying both normal and attack data. This attack data consists of mainly TCP injection, UDP flooding and ICMP flooding attacks. The most important fields of a flow record are the total number of packets in a flow and the total number of bytes in each flow. This combination of attributes of the flow helps in detecting anomalies in the total amount of traffic. Another combination of attributes in the flow to detect the anomalies are the sources and destination IPs and port pairs which provide input to detect the port scans. Table 1 shows the attributes of packets and flows contained in the datasets.

Table 1: Input Data Attributes

Dataset	TCP Packets	UDP Packets	ICMP Packets	Total Flows
Dataset 1	3749	1185	220.5	5154
Dataset 2	1616	1610	345	3570
Dataset 3	2388	657	64.5	3110
Dataset 4	11034	1224	231	12489

https://sites.google.com/site/ijcsis/
ISSN 1947-5500

4. ANALYSIS

The performance metrics are evaluated for the two approaches and a comparative study is presented

Table 2: Baseline characteristics results.

Dataset	Normal Traffic	Abnormal Traffic
Dataset 1	55.31%	44.69%
Dataset 2	58.50%	41.50%
Dataset 3	53.39%	46.61%
Dataset 4	55.60%	44.40%

Results of Experiment– K-Means Evaluation

The data captured in the four datasets as mentioned in table 2 was the input data for the k-means algorithm. k-means algorithm clustered the traffic into normal or abnormal flows. Table 3 shows the percentage of the traffic clustered into normal or abnormal flows for each dataset using k-means. The study plotted the normal and abnormal clusters for each dataset using k-means as shown in figure 1-4. Each flow record of both the clusters are marked as green for normal flow and red for abnormal flow as shown in figures 1-4.

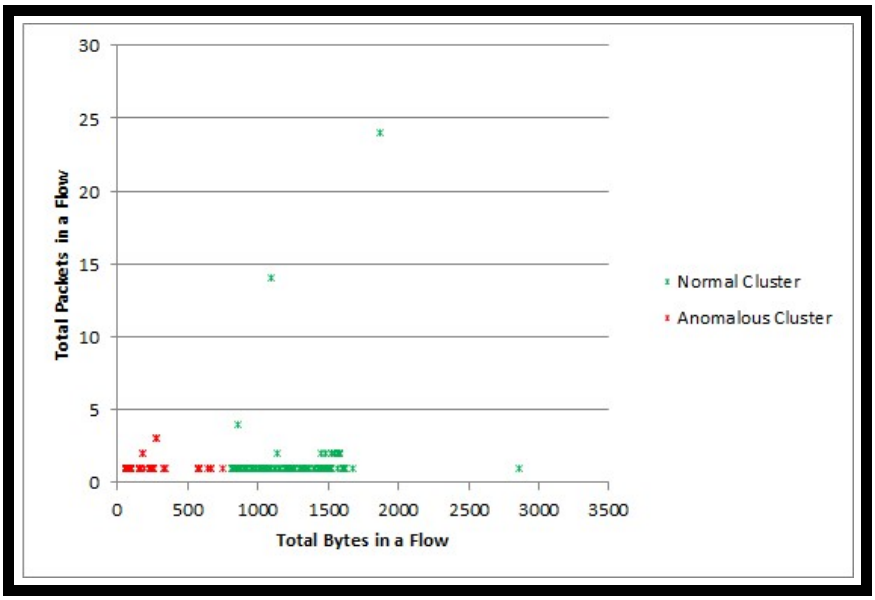

Figure 1: k-means clustering on dataset 1.

https://sites.google.com/site/ijcsis/
ISSN 1947-5500

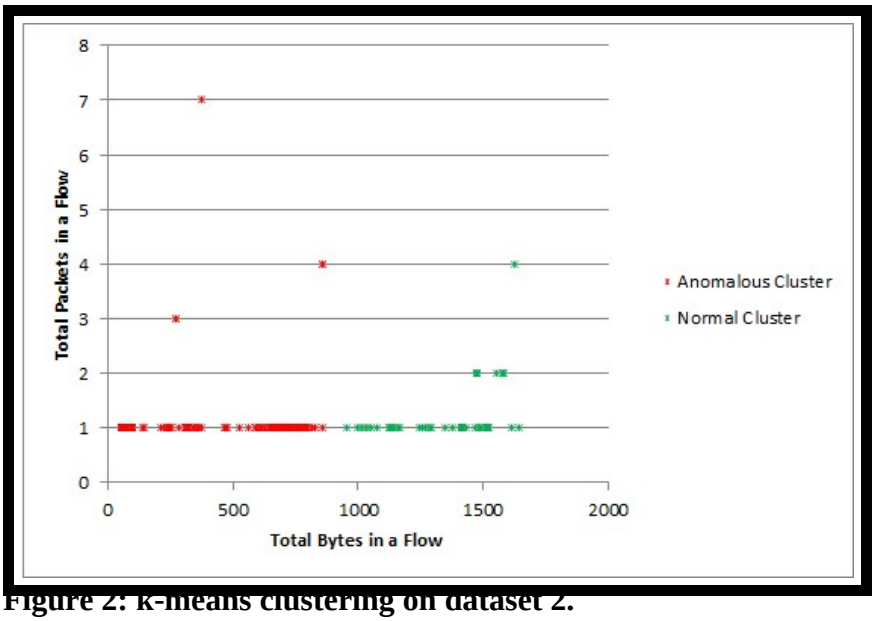

Figure 2: k-means clustering on dataset 2.

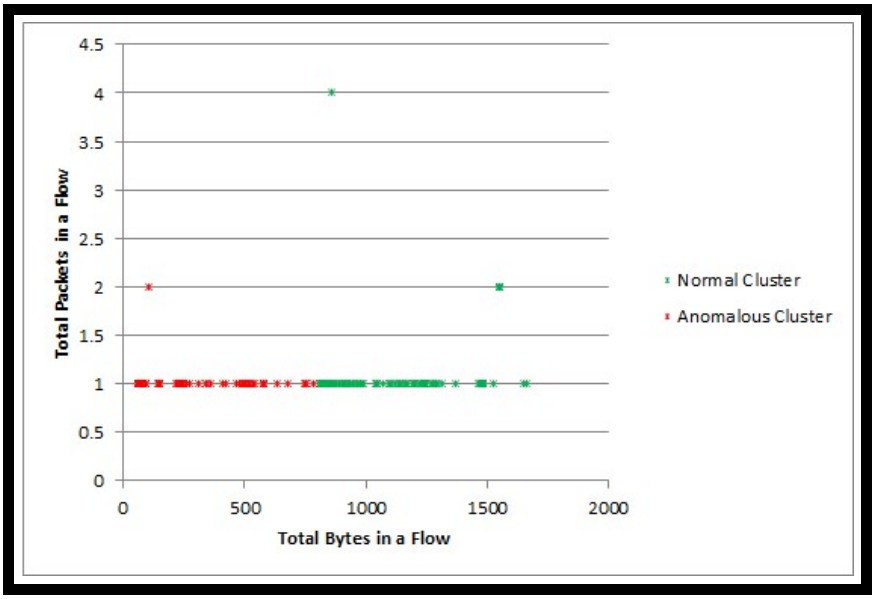

Figure 3: k-means clustering on dataset 3.

https://sites.google.com/site/ijcsis/
ISSN 1947-5500

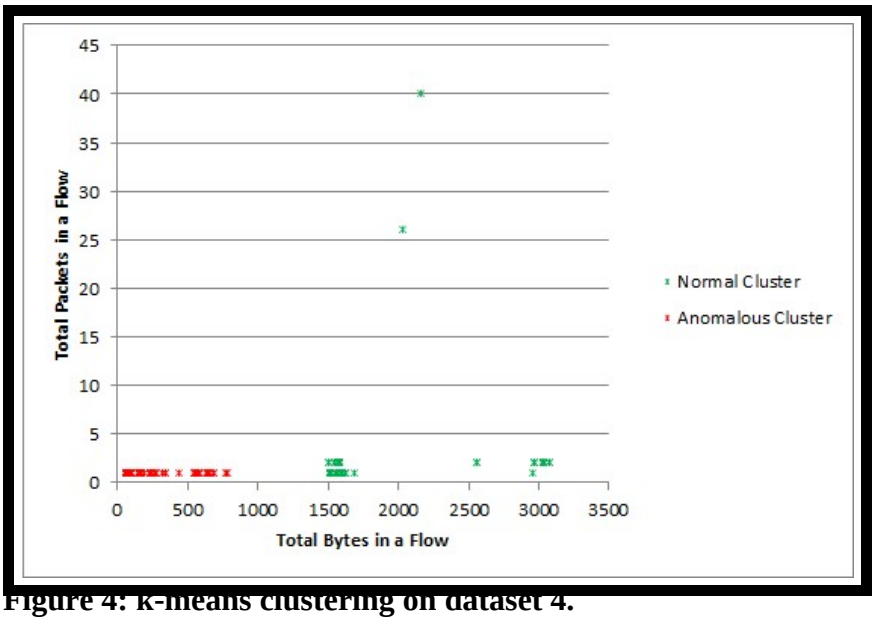

Figure 4: k-means clustering on dataset 4.

The results presented in table 3 shows that for the dataset 1 which consists of mainly TCP injection attacks in the form of several flows having less number of packets and bytes, k-means has assigned 69.35% flows into the abnormal cluster and 30.65% to the normal cluster. In comparison to the traffic characteristics of table 2, the abnormal cluster contains nearly 25% of the normal flows.

The analysis of dataset 2 reveals that k-means have assigned 75.02% flows into the abnormal cluster and 24.98% to the normal cluster. When compared to the baseline traffic characteristics of table 2, the abnormal cluster contains nearly 34% of the normal flows. The analysis of dataset 3 revealed the same characteristics of the k-means algorithm where it has clustered 70.46% flows into the abnormal cluster and 29.54% to the normal cluster. As compared with the traffic characteristics of table 2, the abnormal cluster contains nearly 36% of the normal flows. However, on dataset 4 it was found that k-means was able to cluster 55.55% flows into the abnormal cluster and 44.45% to the normal cluster. k-means has shown improvement in clustering more amount of normal flows into the normal cluster for this dataset. This abnormal cluster contains nearly 11% of the normal flows which is least when compared to other three datasets. This dataset contains highest TCP injection attacks with moderate UDP flooding attacks.

Table 3: Traffic clustering using k-means.

Dataset	Normal Traffic	Abnormal Traffic
Dataset 1	30.65%	69.35%
Dataset 2	24.98%	75.02%
Dataset 3	29.54%	70.46%
Dataset 4	44.45%	55.55%

https://sites.google.com/site/ijcsis/
ISSN 1947-5500

It could, therefore, be concluded that k-means was able to cluster those TCP flows as abnormal which exhibited the similar type of behaviour in terms of less number of packets and bytes.

Results of Experiment – Outlier detection evaluation

The study analysed the results of outlier detection as shown in table 4 for the input datasets and compared them with the baseline behaviour characteristics results of table 2. For dataset 1, outlier detection was able to classify 28.06% out of 44.69% of the abnormal flows as abnormal. For normal flows, 71.94% in excess to 55.31% were assigned score <=1.2. In this case, nearly 15% of the normal traffic was given an outlier score greater than 1.2 and were declared as abnormal. For dataset 2, 20.24% out of 41.5% of the abnormal flows were declared as abnormal. For normal flows, 79.76% in excess to 58.5% were assigned score <=1.2. In this case, again nearly 15% of the normal traffic was given an outlier score greater than 1.2 and were declared as abnormal. The same type of figures exists for dataset 3 where 21.92% out of 46.61% of the abnormal flows were declared as abnormal and 78.08% in excess to 53.39% were assigned score <=1.2 and declared normal. Here also, nearly 15% of the abnormal flows were classified as normal. In dataset 4 which contains many TCP injection traffic, outlier detection has performed badly and classified only 12.07% out of 44.4% of the abnormal traffic as abnormal. The remaining was classified as normal.

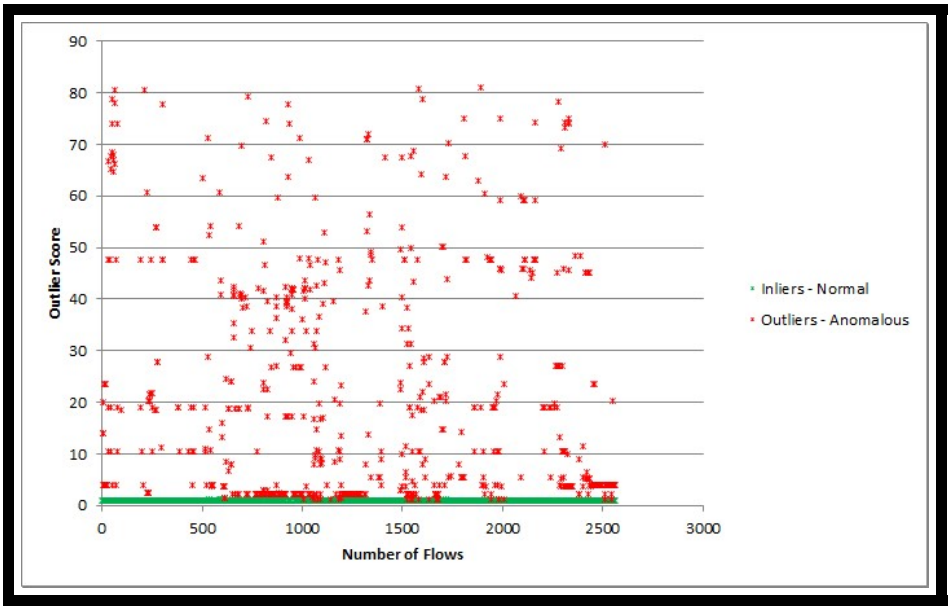

Figure 5: Outliers scores of datasets 1.

https://sites.google.com/site/ijcsis/
ISSN 1947-5500

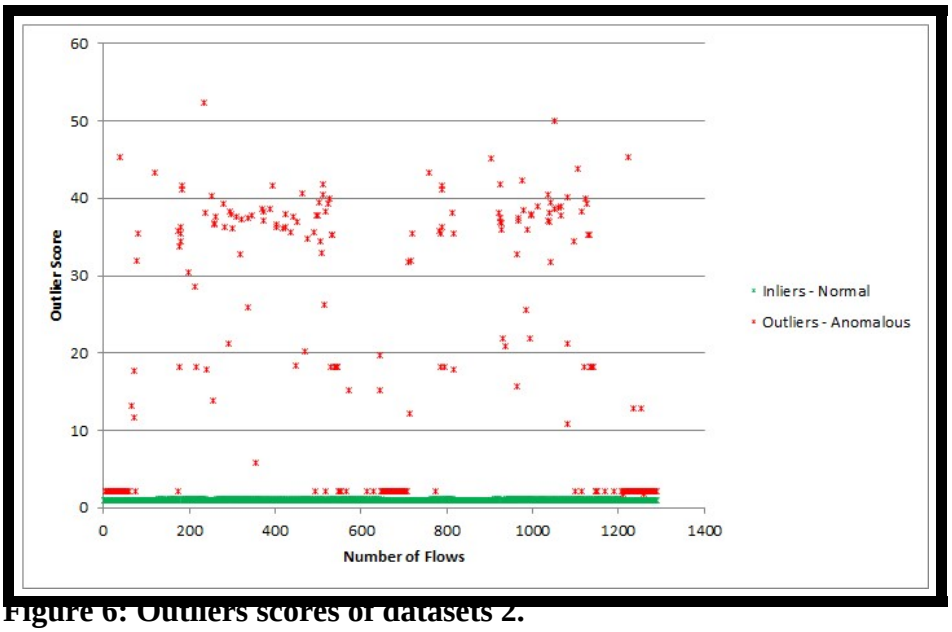

Figure 6: Outliers scores of datasets 2.

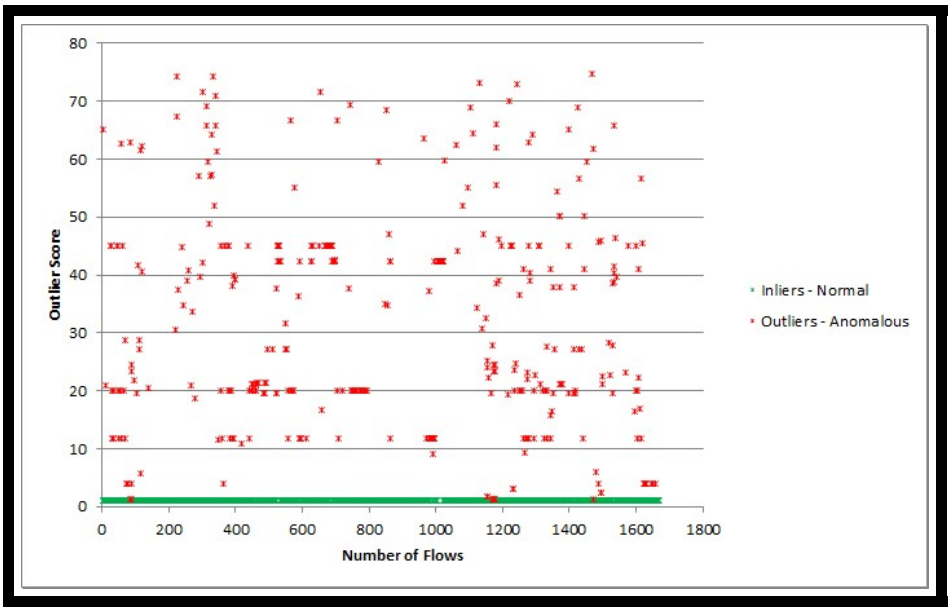

Figure 7: Outliers scores of datasets 3.

https://sites.google.com/site/ijcsis/
ISSN 1947-5500

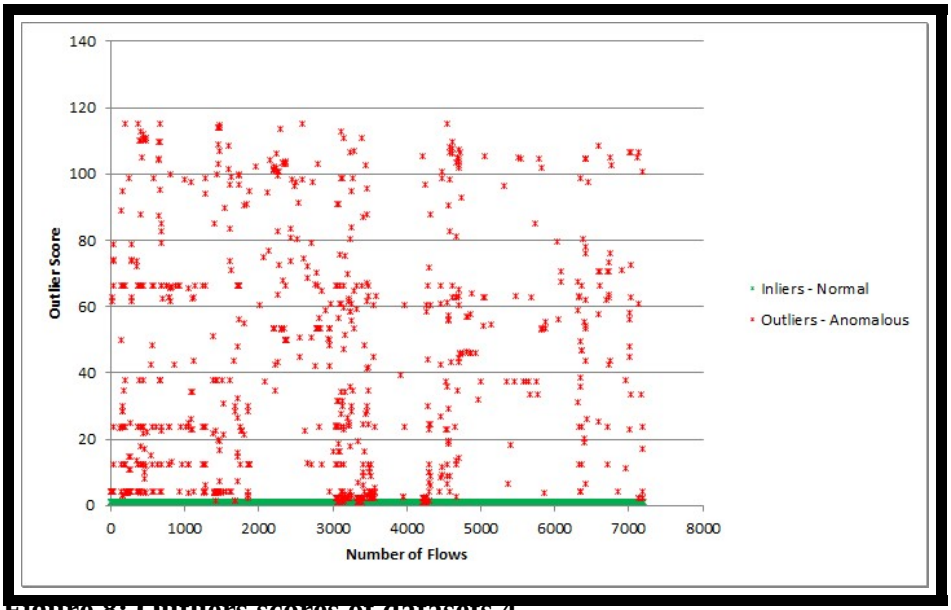

Figure 8: Outliers scores of datasets 4.

Table 4: Traffic classification using outlier detection.

Dataset	Normal Traffic	Abnormal Traffic
Dataset 1	71.94%	28.06%
Dataset 2	79.76%	20.24%
Dataset 3	78.08%	21.92%
Dataset 4	87.93%	12.07%

The following type of abnormal traffic was declared as abnormal by outlier detection.

- ICMP flows originating from different sources to the same destination. (ICMP flooding)
- ICMP flows originating from same sources to the different destination. (ICMP flooding)
- TCP flows originating from different sources and same port to the same destination IP and port and having more than 100 bytes per flow.
- TCP flows originating from same sources to different destination IP and same port and having more than 100 bytes per flow.

The following type of abnormal traffic was declared as normal by outlier detection.

- UDP flows containing more than 500 packets and 10000 bytes per flow. (UDP flooding)
- The following type of normal traffic was declared as abnormal by outlier detection.
- UDP flows originating from different sources to the same destination IP and port.

The study can conclude from the results of outlier detection that outlier detection is better in detecting ICMP flooding. However, it was not able to detect TCP injection where the number of bytes per flow is less than 100 which is generally the case. But in the case of TCP flows where the total bytes exceed 100, it was able to classify them as abnormal which is not always true. However, it was also not able to detect UDP flooding as abnormal in any case.

https://sites.google.com/site/ijcsis/
ISSN 1947-5500

Performance Evaluation using IDS Metrics

The study evaluated the performance of k-means and outlier detection using performance metrics. The study calculated the metrics for all the datasets for k-means and outlier detection and compared them. These metrics values for both the approaches are shown in table 5. The values for these metrics typically range between 0.0 and 1.0. These values were studied, and the following interpretations were made from these performance metrics.

- **Interpretations for False Positive Rate**

The FPR is the rate with which the IDS categorize normal flows as abnormal flows. According to table 5, it can be seen that for all the datasets, the FPR is higher for k-means and lower for outlier detection. The FPR for outlier detection is half of the FPR for k-means. This is due to the observed facts that k-means has clustered 25-35% of the normal traffic into the abnormal cluster in all the datasets. Thus, it can be interpreted that k-means have high FPR which makes it less effective than outlier detection. In this case, the outlier detection has performed better than k-means. It was also observed that irrespective of the amount and variety of anomalies like TCP injection, UDP flooding and ICMP flooding in the datasets, outlier detection has low false positive rate than k-means.

- **Interpretations for False Negative Rate**

The FNR is the rate with which the IDS categorize abnormal flows as normal flows. A high FNR means that the system is more vulnerable to intrusions. Table 5 shows that k-means have much low FNR than outlier detection for datasets 1, 3 and 4. This means that outlier detection has more tendency to generate no alerts on abnormal flows. The FNR for datasets 1, 3 and 4 is much less in k-means as compared to outlier detection. This means that in the case of TCP injection with less amount of UDP flooding, k-means has less FNR. However, FNR for dataset 2 is slightly more in k-means than outlier detection. Thus, the FNR of k-means in the case of detection of heavy UDP flooding is more or equal to FNR of outlier detection which means that kmeans is slightly more vulnerable to UDP flooding than outlier detection. On an average, outlier detection has high FNR than k-means which make it more vulnerable to anomalies.

Table 5: Performance metrics results.

Metrics	Dataset 1		Dataset 2		Dataset 3		Dataset 4	
	k-means	outlier	k-means	outlier	k-means	outlier	k-means	outlier
FPR	0.4	0.2	0.4	0.2	0.4	0.2	0.2	0.1
FNR	0.04	0.6	0.9	0.8	0.01	0.7	0.003	0.8
Sensitivity	0.9	0.3	0.001	0.1	0.9	0.2	0.9	0.1
Specificity	0.5	0.7	0.5	0.7	0.5	0.7	0.7	0.8
CR	0.7	0.5	0.5	0.3	0.7	0.5	0.8	0.5
PR	0.6	0.5	0.4	0.3	0.6	0.5	0.7	0.4

- **Interpretations for Sensitivity**

https://sites.google.com/site/ijcsis/
ISSN 1947-5500

Sensitivity is also known as the true positive rate (TPR) which is a rate according to which abnormal flows are categorized as abnormal. A more sensitive IDS will also have more FPR. So, there is a trade-off between Sensitivity and FPR. An IDS should not be too much sensitive. If it is too much sensitive, then its FPR is also high. It was observed that k-means is more sensitive than outlier detection for datasets 1, 3 and 4 which contain a high amount of TCP injection. Thus k-means has more tendency to give alerts on abnormal flows as compared to outlier detection in case of TCP injection attacks. As a trade-off, k-means has high FPR than outlier detection. The exception noticed here is for dataset 2 containing the highest amount of UDP flooding where k-means is least sensitive as compared to outlier detection.

- **Interpretations for Specificity**

Specificity also known as the true negative rate (TNR) is the rate with which normal flows are categorized as normal. High specificity means that the IDS is more capable of identifying normal traffic as normal. Table 5 shows that k-means has less specificity than outlier detection. But in the case of large datasets containing more number of TCP injection flows, both k-means and outlier detection has proved to be almost equally capable of identifying normal traffic as normal. Overall on an average, outlier detection is more better in declaring normal flows as normal than k-means.

- **Interpretations for Classification Rate**

The Classification Rate tells the measure of how much accurate the declarations are made by the algorithms. For an IDS, the FPR and FNR should be low with maximum CR. However, it was noticed that k-means is more accurate than outlier detection for all the datasets. Thus k-means is more accurate in identifying more number of TCP injections and UDP flooding flows than outlier detection.

- **Interpretations for Precision Rate**

Precision is also known as the positive predictive value (PPV) which gives the measure of detection of real intrusions in the IDS. More PPV indicates that the algorithm is more capable of detecting abnormal flows. Table 5 shows that k-means is slightly more precise in detecting abnormal flows than outlier detection due to its more PPV. Also, k-means precision increased as compared to outlier detection when the size of the dataset increased.

Performance Evaluation Using Computer Resources

The outcome show that the k-means algorithm consumes 10% to 20% of the CPU and takes approximately 5-10 seconds to execute. On the other hand, outlier detection consumed 50% to 60% of the CPU and takes approximately 40-50 seconds to execute on all the datasets. Figure 9 shows the outcome of this experiment on dataset 1. Similar outcome was noticed for all the datasets and it was found that the outlier detection consumed approximately 5 times more CPU and execution time than k-means.

https://sites.google.com/site/ijcsis/
ISSN 1947-5500

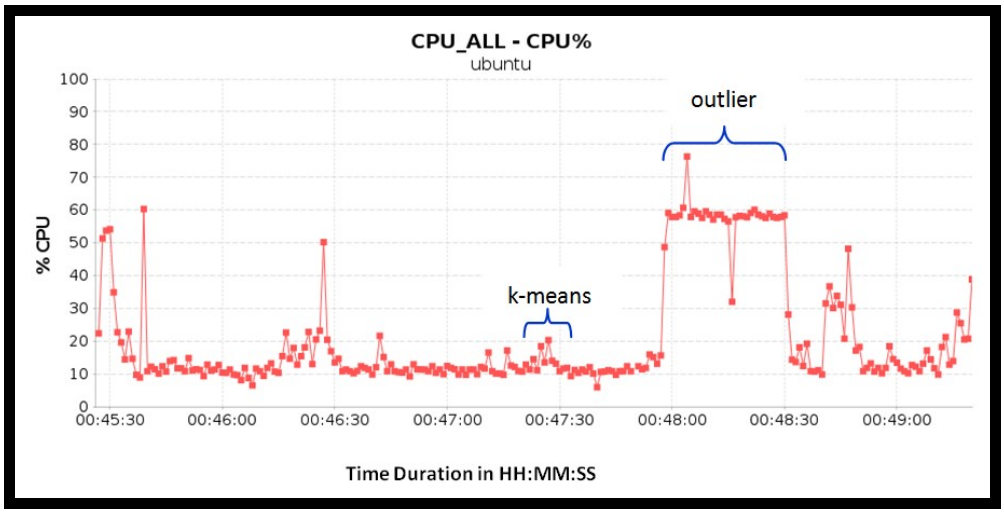

Figure 9: CPU usage on dataset 1.

The study also noticed that the k-means algorithm consumed much less amount of RAM than the outlier detection. Figure 10 shows the amount of RAM freely available during the execution time of the k-means and outlier detection. From this figure, we can infer that around 100 MB of RAM was freely available prior to execution of k-means and outlier detection. The study observed that k-means took an almost negligible amount of RAM as compared to outlier detection which consumed nearly 40% of RAM available freely during its execution. This dropped the free RAM availability to 60 MB. This was observed in all datasets and it was found that the amount of free RAM available during the execution of k-means is 40-45% more than outlier detection.

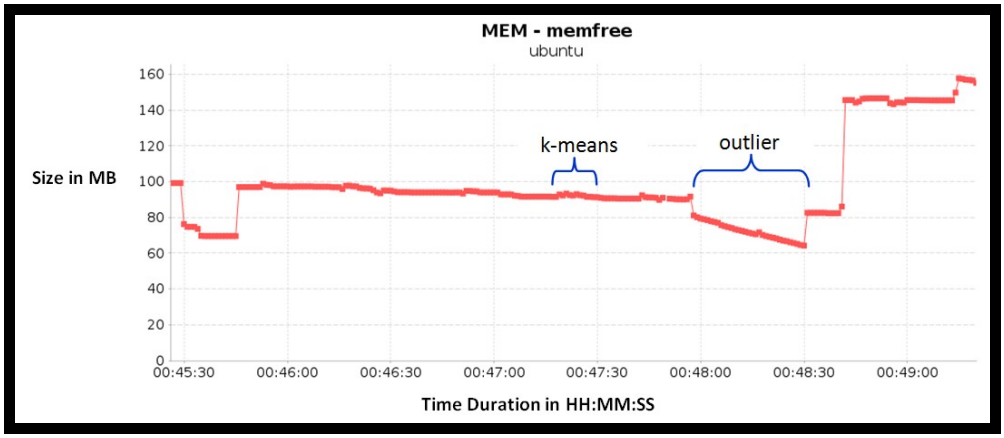

Figure 10: RAM usage on dataset 1.

The study analyzed why outlier detection consumes more CPU and RAM than k-means. The reason for this is that the detection of outliers is significantly involved in the calculation of finding the nearest neighbors of each flow record and that must cross half of the total number of flow records to find the nearest neighbor of each flow record. If n is the total number of flow records, then algorithm for each n must be n/2 iterations. This increases the complexity of outlier detection with NOF. Furthermore, this reduces the efficiency of the outlier detection algorithm with respect to k-means. The complexity of both approaches has been solved. For n flow records in a data set, the assignment of each nth flow record to a normal or abnormal cluster can be done

https://sites.google.com/site/ijcsis/
ISSN 1947-5500

in O (kn) for k-means where k is the number of clusters. However, in the case of outlier detection, finding the nearest neighbors takes n/2 iterations for every nth record that brings its complexity to O (n2).

5. CONCLUSION AND RECOMMENDATION

K-means was able to cluster heavy UDP flooding as abnormal whereas outlier detection has classified it as normal. K-means was better than outlier detection in clustering TCP injections as abnormal containing less number of packets and bytes per flow. But for TCP injections containing more number of packets and bytes per flow, outlier detection performed better. K-means clustered average 65% traffic as abnormal whereas outlier detection classified average 75% traffic as normal.

The study compared the two approaches based on the performance metrics and an amount of computer resources consumed. Outlier detection has low FPR than k-means but proved to be more vulnerable to intrusions than k-means due to the high FNR. However, k-means was more sensitive to outlier detection in generating alerts on abnormal flows. Overall, k-means was more accurate and precise and has better classification rate than outlier detection. Also, the amount of CPU and RAM consumed by outlier detection is much more than k-means which make outlier detection more time expensive.

6.0 FUTURE WORK

The future work includes the optimization of k-means and outlier detection approaches in such a way to detect TCP injections, UDP flooding, ICMP flooding and other DOS attacks with no restriction on the number of packets and bytes per flow. This should be achieved with high classification rate and low false alarm rate. However, the fusion of k-means and outlier detection approaches may help to achieve this, wherein, the k-means would run first to cluster the traffic into normal and abnormal clusters and then calculating the scores for the flows in the abnormal clusters to separate out the normal flows from it. This is because of k-means groups more traffic as abnormal than outlier detection. This may decrease the overall false alarm rate of the IDS and the outlier detection algorithm will also have less number of flow records to traverse through, thereby, consuming less amount of CPU and RAM with fast results. The future work also includes to regularly train the IDS with newer baseline characteristics of normal and abnormal traffic so as to detect newer types of anomalies.

7.0 REFERENCES

[1] Shah A, Waqas J. Rana, “Performance Analysis of RIP and OSPF in Network Using OPNET”, International Journal of Computer Science Issues, Issue 6, No 2, November 2013.
[2] Lammle, Todd, “CCNA Cisco Certified Network Associate study guide, sixth edition”. Indianapolis, Ind.: Wiley. (2007).
[3] Todorovic Ivana, Sepanovic Stevan, “Measurements of convergence time for RIP And EIGRP Protocols”, Scripta Scientiarum Naturalium, volume 2, 2011.
[4] Deng Justice, Wu Siheng, Sun Kenny, “Comparison of RIP, OSPF and EIGRP Routing Protocols based on OPNET” Final project. Spring, 2014.

https://sites.google.com/site/ijcsis/
ISSN 1947-5500

[5] Panford, J. K., Riverson K. & Boansi O. K (2015). Comparative analysis of convergence times between rip, and eigrp routing protocols in a network. Research Journali's; Journal of Computer Science., pg 1-5.

https://sites.google.com/site/ijcsis/
ISSN 1947-5500